Multi-level post-processing for Korean character recognition using morphological analysis and linguistic evaluation[*]

Geunbae Lee, Jong-Hyeok Lee, JinHee Yoo

Department of Computer Science & Engineering

Pohang University of Science & Technology

San 31, Hoja-Dong, Pohang, 790-784, Korea

gblee@vision.postech.ac.kr, jhlee@vision.postech.ac.kr

Summary

Optical character recognition has been actively researched as convenient means of automatic data input to computers. However, due to excessive similarities among recognized characters and noises in images, there have been limitations to direct performance improvements of character recognition method. So post-processing is always required for practical character recognition. Previous post-processing methods use only within-word contextual information such as character transition and confusion probabilities. In contrast, we extend the concept of contextual information to the sentence level, and present a multi-level post-processing method that utilizes linguistic information including character, word, syntax, and even semantic-based knowledge for domain-independent off-line text recognition. The proposed post-processing system performs three-level processing: candidate character-set selection, candidate *eojeol* (Korean word) generation through morphological analysis, and final single *eojeol*-sequence selection by high-level linguistic evaluation. The candidate selection restricts the number of candidates for later processing, and supplements the candidate sets by adding similar characters for error correction. The morphological analysis uses word-fragment level constraints to filter out erroneous recognition results by checking if the recognized character sequences can form a grammatically correct *eojeol*. The linguistic evaluation uses syntax and semantic-level statistical information to further filter out erroneous results. We utilized two high-level linguistic constraints for our linguistic evaluation: tri-gram part-of-speech tagging and mutual information based co-occurrence relations. All the required linguistic information and probabilities are automatically acquired from a statistical corpus analysis. Experimental results demonstrate the effectiveness of our method, yielding error-correction rate of 80.46%, and improved recognition rate of 95.53% from before-post-processing rate 71.2% for single best-solution selection.

Abstract

Most of the post-processing methods for character recognition rely on contextual information of character and word-fragment levels. However, due to linguistic characteristics of Korean, such low-level information alone is not sufficient for high-quality character-recognition applications, and we need much higher-level contextual information to improve the recognition results. This paper presents a domain independent post-processing technique that utilizes multi-level morphological, syntactic, and semantic information as well as character-level information. The proposed post-processing system performs three-level processing: candidate character-set selection, candidate *eojeol* (Korean word) generation through morphological analysis, and final single *eojeol*-sequence selection by linguistic evaluation. All the required linguistic information and probabilities are automatically acquired from a statistical corpus analysis. Experimental results demonstrate the effectiveness of our method, yielding error correction rate of 80.46%, and improved recognition rate of 95.53% from before-post-processing rate 71.2% for single best-solution selection.

Keywords: Korean character recognition, post-processing, morphological analysis, part-of-speech tagging, co-occurrence patterns, linguistic evaluation

1 Introduction

Optical character recognition has been actively researched as convenient means of automatic data input to computers. However, due to the similarities among recognized characters and noises in images, there have been limitations to the performance improvement of character recognition method. Since humans can understand noise-contained images using lexical and grammatical knowledge, character recognition systems also must utilize the contextual information via post-processing of recognized characters. Post-processing can improve the overall recognition performance by correcting the recognition errors and selecting the most appropriate characters among the several candidates according to the given contexts.

Previous post-processing methods [17] use only within-word contextual information such as character transition and confusion probabilities based on Markov assumptions to perform Viterbi-style searches. They also use character similarity metrics to do some dictionary search, and some systems use morphological analysis to find the word structures. However, all these systems have limitations

that they only use within-word contextual information, and do not utilize between-word/phrase information. In contrast, we extend the contextual information to sentence level, and present a multi-level post-processing method that utilizes linguistic information including character[1], word[2], syntax, and even semantic-based information for domain-independent off-line text recognition. The proposed post-processing system performs three-level processing: candidate character-set selection, candidate *eojeol* (Korean word) generation through morphological analysis, and final single *eojeol*-sequence selection by linguistic evaluation. All the required linguistic information and probabilities are automatically acquired from a statistical corpus analysis.

The paper is organized as follows. Section 2 surveys previous approaches for post-processing and their limitations. In section 3, we introduce the high-level linguistic information employed for our post-processing method. Section 4 shows the architecture of the system, and explains the multi-level post-processing method in detail. Section 5 demonstrates the effectiveness of our method by showing several experimental results and analyses, and finally section 6 draws some conclusions.

2 Related researches

Previous post-processing methods mostly utilized character-level contextual knowledge. According to the contextual knowledge representation, these post-processing methods can be classified as bottom-up (data-driven), top-down (knowledge-driven), and bottom-up/top-down hybrid approaches. In the bottom-up methods such as Viterbi algorithm or modified Viterbi algorithm [13, 16], the contextual knowledge is represented probabilistically using Bayesian formalism and Markov assumptions. The algorithm searches for the most-likely solution character sequences given the recognized characters using prior and conditional (confusion) probabilities. The Viterbi algorithm is efficient, but can generate solutions which are not in the given dictionary, which yields relatively low error-correction performance. The top-down methods directly search the dictionary

---

[1]The character in Korean character recognition actually designates a syllable in linguistic terminology. Korean character recognition is performed on syllable-based, rather than alphabet-based as in English, because Korean writing system enforces a two-dimensional syllable structure. When we mention a character regarding to our system in this paper, the readers should know that we actually mean a syllable.

[2]The Korean word is a group of clearly distinguishable morphemes, and is called an *eojeol*. Korean is an agglutinative language which has very complex word structure. In this paper, we will interchangeably use the term 'word' and *eojeol*.

to find the most similar character sequences given the recognized sequences [18]. The dictionary search method usually guarantees good error-correction performance, but also suffers from high costs. The dictionary can be approximated using the binary n-gram (BNA) technique [19]. The BNA dictionary can be used to find if the recognized word contains errors, and also the position of the errors. The BNA technique can also correct the errors, and is more efficient than the direct dictionary search. However, BNA performance is degraded when the word length is short, and the technique generates too many correction candidates. To overcome the limitations of both top-down and bottom-up methods, some hybrid methods are also suggested [12]. These methods basically try to exploit both the efficiency of Viterbi search and the performance of dictionary look-ups. All these previous researches for English language try to find the best solution-character sequences using the character-level information, and rarely try to utilize the more high-level linguistic constraints. However, Korean is an agglutinative language which has very complex word structure, and has two-dimensional syllable-based writing systems. So all these character-based error-correction schemes are too narrow scoped, and cannot give a good performance since Korean recognition should be syllable-based, rather than character-based.

Considering these characteristics of Korean, some researches on Korean character recognition have used morphological analysis and various kinds of linguistic assessments. Lee et. al. [9] used several dictionaries and morphological analysis techniques to correct Korean spelling errors. Their dictionaries consist of morpheme dictionaries and inverse dictionaries of functional words (noun-endings and verb-endings). Later, they extended their methods to incorporate various linguistic heuristics to develop error-type decision functions, and obtained 77.5% of error-correction rate [14]. However, they didn't use any statistical information and solely depended on the symbolic heuristics, therefore yielding error-prone and fragile systems. Hong et. al. [6] used morphological analysis and binary n-gram (BNA) techniques for detecting and correcting errors. Their method showed great efficiency in correcting mis-recognized and un-recognized characters, but the BNA techniques are inherently weak in short-word error correction. Moreover, they couldn't correct the multiple errors occurring simultaneously in two or more morphemes. Lee et. al. [7] argue that post-processing results should be fed-back to the feature extraction and recognition stage. By applying syntactic word structures and character-level probabilities back to the previous stages, they could increase their recognition rate 11% from 86% to 97%. But the feedback can increase

the system complexity and therefore tends to be more time consuming. Due to the similar linguistic structures, morphological analysis and linguistic evaluation have also been used in Japanese character recognition post-processing. In [11], they used morphological analysis to produce all the possible candidate strings and applied evaluation functions based-on Japanese word- or phrase-level heuristics to calculate the phrase plausibilities. By using the evaluation functions, they could increase their recognition rate 6.8% in average. Some systems used detailed domain knowledge to the error-correction and showed a great success. For example, Lee and Kim [8] used special dictionaries and algorithms designed for each of province names, address numbers, building names, and people names in postal addresses, and obtained very good performance of error correction. Similarly, [10] also utilized domain knowledge as well as linguistic knowledge to evaluate the plausibility of bunsetsu (Japanese word) candidates, and could improve the recognition rate for even very un-reliable recognition devices. However, these systems are domain-dependent and cannot be compared with the general purpose post-processing systems.

Contrary to English systems which mostly use character and word-fragment level information, our post-processing scheme focuses on beyond morpheme and between *eojeol* linguistic constraints for more broad and efficient error-correction for Korean. Unlike the previous Korean systems, our scheme utilizes both statistical and symbolic information for efficient error-correction, and employs multi-level feed-forward architecture incorporating all the character-level, morphological, syntactic and semantic co-occurrence knowledge. Each of the knowledge is used in domain-independent way, so our scheme can be well applied to general texts regardless of their domains.

3 High-level linguistic information for post-processing

Broadly speaking, the linguistic information used in post-processing can be any kind of statistical or structural linguistic constraints from character level to semantic level, or even to pragmatic level. The followings are summary of linguistic constraints that can be utilized in character recognition post-processing:

- character/word-fragment level: character confusion probabilities, character transition probabilities, and character-based n-grams

- morpheme/word level: word structure information (morphotactics) and lexical frequencies

- syntax level: structural or statistical relations between words/phrases including part-of-speech tags

- semantic level: semantic selectional restrictions, and word co-occurrence relations

Our post-processing extends the linguistic information up to the semantic level for practical post-processing performance, especially for off-line printed character recognition for massive texts. This section explains the high-level linguistic information (syntax and semantics level) for the post-processing. These linguistic constraints provide the basis for linguistic evaluation during the multi-level post-processing.

3.1 Part-of-speech tags and tagging

A single word can usually have multiple part-of-speech's (POS's) according to the given contexts, and when it is the case, we say that the word exhibits a POS ambiguity. POS tagging is a disambiguation process that assigns the most appropriate POS tag sequence to a given sentence (word sequence) by utilizing the contextual information. When the character recognition results give several possible morphological analyses, the POS tagging can provide syntax-level constraints in order to delete erroneous recognition results. In this paper, we employ the tri-gram tagging model based on HMM (hidden-markov model) process [4]. Constructing an appropriate tagset is essential for any tagging application, and usually the tagset must be in the proper granularity. Extremely refined tagset promises the best application performance, but the tagset tends to be impractical in size. We use a total of 20 tags for morphemes as shown in table 1. Since Korean word (called *eojeol*) usually consists of two or more morphemes, an *eojeol* tag becomes a concatenation of the constituent morpheme tags.

****

Table 1 goes about here

****

The tagging unit can be a morpheme or an *eojeol* in Korean. However, since the morphological analysis already provides the constraints between morphemes, we adopt an *eojeol* as our tagging unit to obtain the necessary syntactic constraints for character recognition post-processing. The tri-gram tagging model computes the best tag sequence $t_{1,n}$ that satisfies the equation (1) in a

7

given sentence. The sentence is composed of morphologically analyzed *eojeol* sequence $e_{1,n}$.

$$T(e_{1,n}) = argmax_{t_{1,n}} p(t_{1,n} \mid e_{1,n}) \tag{1}$$

Using Bayesian reformulation to drop the constant *eojeol* sequence probability, and applying two Markov assumptions to the resulting joint probability that 1) the current *eojeol* only depends on the current tag, and 2) the current tag depends on the previous two tags, equation (1) can be transformed into equation $(2)^3$.

$$T(e_{1,n}) = argmax_{t_{1,n}} \prod_{i=1}^{n} p(e_i \mid t_i) p(t_i \mid t_{i-2,i-1}) \tag{2}$$

In equation (2), $p(e_i \mid t_i)$ and $p(t_i \mid t_{i-2,i-1})$ are called lexical probability and contextual probability respectively, and these probabilities can be estimated by the frequency counts from a corpus as follows:

$$p(e_i \mid t_i) = \frac{freq(e_i, t_i)}{freq(t_i)} \tag{3}$$

$$p(t_i \mid t_{i-2,i-1}) = \frac{freq(t_{i-2,i})}{freq(t_{i-2,i-1})} \tag{4}$$

Using these two frequency count estimations, Viterbi algorithm is applied to search the optimal tag sequence satisfying equation (2) efficiently in polynomial time.

3.2 Co-occurrence patterns

A word which exhibits specific meaning tends to occur in a certain context with other specific words, and the phenomenon is called co-occurrence relations. For example, in Korean, the word ip^4 (mouth) usually occurs with the word *ta-mwul-ta* (shut). Even if the word *ta-mwul-ta* has the meaning of "shut", it cannot occur with the word *mwun* (door) in Korean. The co-occurring word pairs develop co-occurrence patterns, which can give semantic constraints for the recognized words in a sentence. There have been many researches for automatically extracting co-occurrence patterns from a corpus in several application areas [20, 15]. We want to use the co-occurrence patterns as semantic constraints to disambiguate the several candidate *eojeols* in the post-processing.

---

[3]The boundary condition should be considered when $i = 1$ and $i = 2$ in this equation.

[4]Yale romanization is used for Korean alphabets through out in this paper.

There are two types of co-occurrence relations used in our post-processing system. The first relation is between predicates[5] and their nominals, which can be used as the predicate-argument selectional restrictions. We do not use any structural information that requires any form of parsing process to extract the co-occurrence relations. The post-processing mostly needs a lexical disambiguation, rather than a structural one, so the parsing overhead cannot be traded off in the efficient character recognition post-processing. Moreover, current parsing technology is not robust enough to handle unrestricted texts. Instead, we just simply extract the *eojeols* and the part-of-speech's to represent the co-occurrence relations. The second co-occurrence patterns occur between two mutually associated nominals. For example, the word *un-hayng* (bank) usually occurs with the word *ton* (money). In this case, we usually take into account the words which are associated with only limited number of other words. If a word tends to occur with so many other words, then the word is too general to be associated with any specific word, and the co-occurrence patterns become meaningless in this case. The degree of word generality can be calculated using the following generalization factor:

$$generalization\ factor = \frac{the\ number\ of\ co-occurring\ words}{frequency\ of\ the\ word\ itself} \tag{5}$$

We only consider the words with small generalization factor to extract the meaningful co-occurrence patterns.

The co-occurrence relations can be quantified by calculating mutual information among the co-occurring words. The mutual information is an information-theoretic measure of the word association, and can be calculated based on a corpus. The mutual information I(x,y) between two words x and y is defined as in equation (6) [11].

$$I(x,y) = log_2 \frac{p(x,y)}{p(x)p(y)} \approx log_2 \frac{Nf(x,y)}{f(x)f(y)} \tag{6}$$

In equation (6), p(x) and p(y) designate word occurring probabilities, and p(x,y) is a joint occurring probability of the two words x and y. The probabilities can be approximated using the word occurring frequencies f(x) and f(y), and the joint occurring frequencies f(x,y) within a sentence, all of which can be acquired from a corpus of size N. The calculated I(x,y) has bigger values when the two words are strongly associated. In that case, the co-occurrence patterns exhibit more strong semantic constraints for the post-processing.

---

[5]In Korean, verb, adjective and verbalized nouns (noun + predicate-particle) are used as predicates.

4   Multi-level post-processing

Basic purpose of the post-processing is a disambiguation of multiple recognition results. In our system, input to the post-processing is a recognition result which consists of (candidate, distance) pairs for each character (Korean syllable). The distance is a normalized recognition score between an input pattern and its candidate pattern, and becomes smaller when the recognition accuracy gets higher. Among the recognized candidates, the post-processing selects the best candidate character in a given context by applying multi-level constraints in order to delete the inappropriate recognition results. Applying multi-level constraints is especially necessary for Korean character recognition because Korean recognition is syllable-based, not single character-based such as in English recognition. If we only apply character-level probabilistic information, we cannot cope with the complex word structures. The Korean dictionary needs a morpheme as a header so the probabilistic dictionary look-up for closest word match is not efficient because it requires word-based dictionary search. We adopt a multiple filtering scheme that selects the final solutions step-by-step among all the possible candidates. Figure 1 shows our multi-level post-processing architecture.

****

Figure 1 goes about here

****

The candidate selection uses character-level information to restrict the number of candidates for simplicity of later processing, and also to supplement the candidate sets by adding similar characters for error correction. The morphological analysis uses word-fragment level constraints to filter out erroneous recognition results by checking if the recognized character sequences can form grammatically correct *eojeols*. The linguistic evaluation uses syntax and semantic-level statistical information to further filter out the erroneous results.

4.1   Candidate selection

4.1.1   Candidate restriction

The recognition device produces many candidates for each character, so the character combinations can exponentially increase in a word. The excessive number of candidates increases the post-processing time and decreases the overall recognition rates due to excessive false alarms in the

dictionary look-up. The candidate restriction is performed based on the recognition score of the best scored candidate (called the first candidate) for each character. If the score of the first candidate is very high, then many candidates can be curtailed safely because the character is well-recognized in this case. To formulate the candidate restriction process, suppose S_0 is a set of (candidate, distance) pairs for a character, sorted by increasing order of the distance.

$$S_0 = \{(c_1, d_1), (c_2, d_2), \ldots (c_n, d_n)\} \tag{7}$$

where c_i and d_i are the i-th candidate and distance. The result of the candidate restriction can be represented in S_1:

$$S_1 = \{(c_i, d_i) \mid (c_i, d_i) \in S_0, d_i - d_1 < \theta_1, \frac{d_i - d_1}{d_1} < \theta_2\} \tag{8}$$

where θ_1 and θ_2 are thresholds of the restriction which should be determined to reflect the characteristics of the recognition device.

4.1.2 Candidate supplement

The candidate supplement is required for very similar characters which are almost impossible to be distinguished by using only the pattern themselves. Especially, Korean has a lot of similar characters that result in frequent recognition errors [5]. For each mis-recognizable character, candidate supplement recovers recognition errors by inserting its similar characters into its candidate set. We use the similar-character table for Korean in which mutually mis-recognizable characters are collected in pairs. The similarity between characters was determined by the experiments [5]. The candidate supplement process can be formulated as follows:

$$S_2 = S_1 \cup \{(c, d_i) \mid (c_i, d_i) \in S_1, \ (c_i, c) \in similar - character \ table\} \tag{9}$$

To prohibit the excessive increases in candidate numbers, currently we only supplement the first candidates that have the minimum distance. Figure 2 shows the output of the recognition device that produces 10 candidates for each recognized character.

****

Figure 2 goes about here

****

After performing the candidate restriction and supplement, the candidate set is like in figure 3.

****

Figure 3 goes about here

****

4.2   Morphological analysis

The morphological analysis segments an *eojeol* into a sequence of morphemes, and recognizes the constituent morphemes' root forms from phonological changes. Usually many morpheme combinations are possible in a single *eojeol*, so we must have the knowledge of morphotactics to extract only grammatically correct morpheme combinations. The morphological analysis also must handle the phonological changes such as irregular conjugation, hiatus, contraction, and so on. The morphological analysis can play important roles in character recognition post-processing since it can filter out erroneous recognition results by checking if the sequence of recognized characters can form a grammatically correct combination of morphemes (that is, an *eojeol*). We developed an Korean morphological analyzer based on a tabular parsing method [2]. The algorithm utilizes two linguistic resources: a trie-structured morpheme dictionary and a connectivity-information table. The dictionary encodes the hierarchically organized and morpho-syntactically refined part-of-speech (POS) symbols[6] for each morpheme entry, and the connectivity-information table encodes all the possible combinations between these POS symbols. The morphological analysis should be performed on every sequence of characters that can be formed by permutations of each recognition candidate. However, since the number of possible sequences grows exponentially, we organize the dictionary in the trie structure [1] to utilize the trie's prefix-closed property, that is, if a string is in a trie, then all the prefixes of the string must also be in the trie. Since our morphological analysis is performed by scanning from right to left, the trie actually contains reverse strings of the morphemes.

The morphological analysis based on the tabular parsing consists of two important processes: dictionary search and connectivity checking (see figure 4). The dictionary search extracts all the possible morphemes in an *eojeol*, and the connectivity checking deletes out all the grammatically

---

[6]From the basic part-of-speech, we developed very fine grained categorization of every Korean morpheme (about 400 categories). These 400 fine grained category symbols are used for the Korean morphotactics modeling. Note that the tag-set used in POS tagging is a subset of these 400 category symbols.

incorrect morpheme combinations.

****

Figure 4 goes about here

****

The dictionary search position is controlled using the triangular-table where T[i,j] holds the morphological analysis results between i-th and j-th character in an *eojeol*. The T[i,j] can be formed either by a single morpheme or by a combination of morphemes in the T[i,k] and T[k+1,j], where k is between i and j-1. So the algorithm is in principle a dynamic programming technique. Figure 4 shows the description of the algorithm, and figure 5 shows a example morphological analysis result in the triangular-table. Since all the partial results (intermediate combinations of the morphemes) are in the position of the last column, the actual time complexity is $O(n^2)$ at worst case when n is the number of characters in the input *eojeol*. However, since the trie property can access all the prefixes of the found string at once, the actual dictionary access time is $O(n)$.

****

Figure 5 goes about here

****

4.3   Linguistic evaluation

The morphological analysis usually selects several morphologically-correct *eojeols* in a sentence, but not all of them are correct in the given syntactic and semantic contexts. As the final level of post-processing, we score each *eojeol* according to the high-level linguistic constraints, and select a single correct *eojeol* depending on the scores. The high-level linguistic constraints used are syntactic-level tagging scores and semantic-level co-occurrence scores. Since the tagging and the co-occurrence relations are already explained in section 3, this section only illustrates how the two linguistic constraints are actually applied to the post-processing.

Figure 6 shows how tri-gram tagging filters out implausible candidate *eojeol* sequences.

****

Figure 6 goes about here

****

Since the tagging only relies on the syntactic-level constraints that are manifested by the *eojeol*

13

lexical probabilities and transition probabilities, there still remain semantic ambiguities even in the best tagging paths as shown in figure 6 (represented as the solid arrows). For the safe pruning, we select the n-best tagging paths and deliver the multiple results to the semantic co-occurrence checking process. The co-occurrence patterns can help produce further semantically-disambiguated *eojeols* after the tagging process. This process works especially well when the nominals or predicates are in the ambiguous *eojeols*. The mutual information (see section 3.2) for the nominals (or predicates) between in the ambiguous *eojeols* and in the previously disambiguated *eojeols* is calculated, and the best scored *eojeols* can be selected. For example, in figure 6, the mutual information gives the final disambiguated results *tte* and *pwul-ey* at the 6th and 9th *eojeol* positions among the still ambiguous results (designated by the light dark circles). Even after the high-level linguistic evaluation, there is a chance that the ambiguity still remains. In that case, we select the final *eojeol* that has the smallest distance sum according to the candidate order from the recognition device.

5 Experiments

5.1 Experiment set-up

The experiment set-up for the multi-level post-processing is shown in figure 7. The original texts, recognized texts, and the post-processed texts are compared one another to obtain the recognition rate and the correction rate.

****

Figure 7 goes about here

****

For the post-processing experiments, the following resources have been prepared:

- dictionary: a trie-structured dictionary with about 30,000 morphemes, and a connectivity information table.

- similar character table: character (syllable) similarity is calculated from each phoneme (consonant and vowel) similarity and the recognition device confusion probabilities. We constructed about 100 entries of similar character table for Korean.

- tagged corpus: lexical and transition probabilities for tagging, and mutual information for

co-occurrence patterns are acquired from a tagged corpus. We built a tagged corpus using about 3,000 sentences (23,000 *eojeols*) from elementary-school textbooks and raw sentences supplied from ETRI[7]. From this tagged corpus, we extracted about 140 uni-grams, 1,300 bi-grams, and 5,000 tri-grams for *eojeol* tags.

- test data: the 1,722 *eojeol* test data are selected from the elementary-school textbooks, and divided into 3 sets A, B, C according to the OCR recognition rate (68.4 %, 69.5 %, 75.6 % respectively).

5.2 Experiment results and analyses

5.2.1 Performance measures

We use correction rate and recognition rate (after post-processing) for our performance measures. The correction rate is defined as follows:

$$correction\ rate = \frac{(successfully\ corrected\ characters) - (mis-corrected\ characters)}{total\ erroneous\ first\ candidates} \times 100$$
(10)

where "mis-corrected" means that the correctly recognized characters become incorrect due to the post-processing. On the other hand, the recognition rate is defined as follows:

$$recognition\ rate = \frac{correctly\ recognized\ characters}{total\ first\ candidates} \times 100$$
(11)

Figure 8 and figure 9 shows the correction rate and the recognition rate (before and after post-processing) for characters and *eojeols* with each document set A, B, C and their average. The correction rate is high when the original recognition rate (before post-processing) is high. This means that the error correction performs well for the highly confident candidate sets that have small distances. However, the overall recognition rate after post-processing generally becomes high even for the low original recognition rate, so the post-processing can be practically used for the low recognition-rate devices.

****

Figure 8 goes about here

****

---

[7]Electronics and telecommunications research institute in Korea

****

Figure 9 goes about here

****

5.2.2   Candidate selection effects

The post-processing is performed on sentences, so the processing time depends on the number of candidate *eojeols* generated by the morphological analysis and the sentence length. The candidate *eojeols* are composed of candidate character combinations, so the processing time exponentially increases according to the number of candidate characters. Too many candidate characters also degrade the recognition rate since the *eojeols* made of low order candidates might get high scores in the linguistic evaluation. However, too few candidates might result in no solution in the candidate character set. Figure 10 shows the effect of candidate restriction by showing the recognition rate according to the threshold θ_1 (see section 4.1.1).

****

Figure 10 goes about here

****

As shown in the figure, the number of candidates that yields the best recognition rate depends on the document sets, hence on the recognition devices. We have to choose the best threshold values according to recognition devices through experiments. The post-processing assumes that there is at least one correct solution in the candidate set. However, in reality, the Korean character set (2350 different characters) contains so many similar characters that there might not be any solution character in the candidate character set. Therefore, we supplemented the first candidate to include all the similar characters according to the device confusion probabilities and the original character similarity. Figure 11 shows the candidate supplement effects.

****

Figure 11 goes about here

****

5.2.3 Ambiguity resolution performance

The post-processing process can be interpreted as a disambiguation process that selects a single solution character among several candidate characters. We apply multi-level linguistic constraints for the disambiguation of characters in an *eojeol* structure. Figure 12 shows a disambiguation performance of each linguistic constraint application: the morphological analysis, the tagging, and the co-occurrence patterns.

****

Figure 12 goes about here

****

The ambiguity resolution rate for each specific linguistic processing is defined as follows:

$$ambiguity\ resolution\ rate = \frac{recognition\ rate\ increase\ after\ the\ specific\ linguistic\ processing}{total\ recognition\ rate\ increase}$$

(12)

Even after applying all the linguistic constraints, about 4% test data still have ambiguities. So we had to decide the final solutions based on the candidate order from the recognition device.

5.3 Discussions

The linguistic information used is statistical, rather than structural, so it can be automatically extracted from a corpus and is robust in its nature. However, the statistical information inherently depends on the corpus, so the words which are not in the training corpus result in zero frequencies in the post-processing. So some form of smoothing is always necessary to deal with this sparse data problem. For the tri-gram tagging, we used the uni-gram and bi-gram together for the smoothing[3].

$$p(t_i \mid t_{i-2,i-1}) = \lambda_1 p(t_i) + \lambda_2 p(t_i \mid t_{i-1}) + \lambda_3 p(t_i \mid t_{i-2,i-1})$$

(13)

where $\lambda_1 + \lambda_2 + \lambda_3 = 1$. The sparse data problem also generates the zero co-occurrence frequences in mutual information calculation, and results in $-\infty$ in the value. Basically, this problem can be handled with the semantic category-based mutual information using a well-developed thesaurus. However, well-developed Korean thesaurus is not available at the moment, so we had to develop another smoothing technique. In order to cover the words that do not co-occur in the training data, we employ the single word frequencies together with the mutual co-occurrence frequencies such as:

$$I_{new}(x,y) = \lambda_1(f(x) + f(y)) + \lambda_2 I(x,y)$$

(14)

17

Usually the co-occurrence pattern size is enormous when we consider all the (predicate, nominal) and (nominal, nominal) pairs in the dictionary. However, our system only extracts the co-occurrence patterns for the words that occur in the real corpus. Moreover, we only consider the words that have more than a certain amount of actual co-occurrence frequencies, and that have restricted number of accompanying words using the generalization factor (see section 3.2). Our experiments use 22,000 *eojeol* corpus that includes about 2,000 predicates and 900 common nominals. According to our scheme, only about 2,800 co-occurrence patterns were actually extracted among the theoretically possible more than billion word pairs.

6   Conclusions

This paper proposes a practical post-processing system for optical character recognition, which utilizes high-level linguistic information as well as character-level information. Our post-processing method is especially useful for the applications that require beyond word-level contexts to improve the recognition results, such as off-line massive text recognition. Unlike most of the previous post-processing schemes that utilize only character and word-fragment level information, our post-processing is executed in 3 stages: candidate character-set selection, candidate *eojeol* generation through morphological analysis, and final single eojeol-sequence selection by the high-level linguistic evaluation.

The candidate selection uses the distance generated by the recognition device, and restricts the number of candidates for later processing, and supplements the candidate sets by adding similar characters for error correction. For the selected candidate characters, the morphological analysis generates only the morphologically-correct *eojeol* sequences by checking if the recognized character sequences can form a grammatically-correct *eojeol*. The generated *eojeols* are now grammatically correct, but may be inappropriate in the given contexts. The linguistic evaluation uses syntax and semantic-level statistical information to further filter out the contextually-inappropriate *eojeols* for final recognition error correction. The linguistic evaluation is performed in a cascaded way using syntactic tagging constraints, semantic co-occurrence constraints, and finally candidate orders from the recognition device.

We conducted extensive experiments to demonstrate the performance of our multi-level post-

processing method. For the 1,722-*eojeol* test data extracted from elementary-school textbooks, we obtained 80.46% correction rate and 24.3% increase of the recognition rate (from 71.2% to 95.53%). This performance is much better than similar previous approaches for Korean and Japanese post-processing compared in section 2. Moreover, our post-processing can be applied to any text in domain-independent way. The major post-processing failures in our system come from the case that the selected candidate set does not include the solution characters in the first place since our test-bed recognition device is primitive and experimental one. This no solution case propagates to the next stages of the post-processing, resulting in the morphological analysis failures or incorrect *eojeol* selection which again gives rise to the tagging and co-occurrence checking failures. The better recognition devices should yield much better post-processing results as demonstrated in figure 8 and figure 9. The post-processing failures are also due to the limited corpus size which gives incomplete statistical linguistic constraints in the tagging and co-occurrence pattern extraction. The larger-scale balanced corpus should be provided for more practical post-processing.

References

[1] A. V. Aho, J. E. Hopcroft, and J. D. Ullman. *Data structures and algorithms.* Addison-Wesley Publishing Company, 1983.

[2] A.V. Aho and J. D. Ullman. *The theory of parsing, translation, and compiling, Vol 1: parsing.* Prentice-Hall, Englewood Cliffs, NJ, 1972.

[3] E. Charniak. *Statistical language learning.* MIT press, 1994.

[4] K. Church. A stochastic parts program and noun phrase parser for unrestricted text. In *Proceedings of the conference on applied natural language processing*, pages 136–143, 1988.

[5] Y. H. Do, Y. S. Whang, and S. Y. Bang. Similar character sets in Korean. In *Proceedings of the first Korean character recognition workshop*, 1993 (in Korean).

[6] N. Hong, W. Lee, J.H. Lee, and G. Lee. A study on the correction method of misrecognized characters using string and eojeol information in character recognition system. In *Proceedings of the first Korean character recognition workshop*, 1993 (in Korean).

[7] B.H. Lee, I. D. Lee, and T. K. Kim. Error correction using feedback information in the Hangul text recognition system. In *Proceedings of the first Korean character recognition workshop*, 1993 (in Korean).

[8] S. W. Lee and E. S. Kim. Efficient postprocessing algorithms for error correction in handwritten hangul address and human name recognition. *Pattern recognition*, 27(12):1631–1640, 1994.

[9] Y. S. Lee, Y. S. Chae, Y. S. Yoon, and H. C. Kwon. Korean spelling correction system. In *Proceedings of the 93 Korean artificial intelligence workshop*, 1993 (in Korean).

[10] H. Niwa, K. Kayashima, and Y. Shimeki. A postprocessing for character recognition by keyword extraction and optimum bunsetsu selection. *The Tranactions of IEICE (The Institute of Electronics, Information and Communication Engineer)*, J76-D-2(5):940–948, 1993.

[11] H. Sambe and K. Ohmori. Correction of erroneously recognized handwritten characters from low reliable recognition system by using linguistic information. *Transactions of information processing society of Japan*, 34(19):2117–2124, 1993 (in Japanese).

[12] R. Shinghal and G. Toussaint. A bottom-up and top-down approach to using context in text recognition. *Int. J. man-machine studies*, 11:201–212, 1979.

[13] R. Shinghal and G. T. Toussaint. Experiments in text recognition with the modified Viterbi algorithm. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 1:184–193, 1979.

[14] C. M. Sim and H. C. Kwon. Korean spell correction using collocation of morphemes. In *Proceedings of the 6th conference on Korean and Korean information processing*, 1994 (in Korean).

[15] F. Smadja and K. McKeown. Automatically extracting and representing collocations for language generation. In *Proceedings of the 28th annual meeting of the ACL*, 1990.

[16] S. Srihari, J. Hull, and R. Choudari. Integrating diverse knowledge sources in text recognition. *ACM Trans. on office information systems*, 1(1):68–87, 1983.

[17] S. N. Srihari. *Tutorial: Computer text recognition and error correction*. IEEE computer society press, 1985.

[18] G. Toussaint. The use of context in pattern recognition. In *Proc. IEEE computer society conference on pattern recogntion and image processing*, pages 1–10, 1977.

[19] J. Ullman. A binary n-gram technique for automatic correction of substitution, deletion, insertion and reversal error in words. *The computer journal*, 20:141–147, 1977.

[20] M. Yamashina and F. Obashi. Collocation analysis in Japanese text input. In *Proceedings of the twelfth international conference on computational linguistics (coling)*, pages 770–772, 1988.

tag	description	tag	description
MP	proper noun	SC	ordinal numeral
MD	bound noun	SO	cardinal numeral
MC	common noun	e	prefinal ending
D	verb	y	predicate particle
H	adjective	mC	conjunctive ending
G	adnoun	mT	final ending
B	adverb	mJ	derivative ending
jJ	conjunctive particle	T	pronoun
jS	auxiliary particle	+	prefix
jC	case particle	-	suffix

Table 1: Morpheme tagset for Korean part-of-speech tagging.

sequence of

(candidate, distance) pairs

↓

candidate
selection

similar
character
table

↓

morphological
analysis

dictionary

connectivity
information

↓

linguistic
evaulation

tagging probabilities

mutual information

↓

sequence of
solution characters
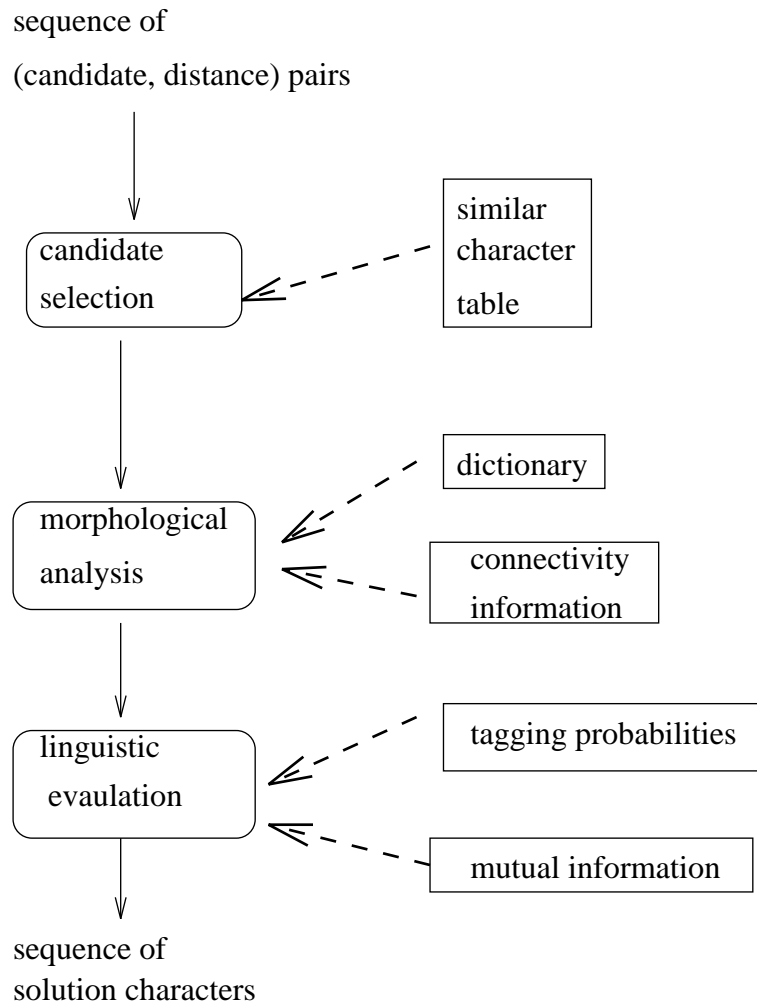
Figure 1: The architecture of multi-level post-processing

```
[cim 104][cam 137][cal 185][kiph 197][chim 205][cap 210][cep 215][kil 227]...
[kyel 31][kel 75][kyeth 92][kil 120][kal 121][cil 135][cel 137][kath 200]...
[ney 181][ey 186][sey 232][ay 245][nay 289][may 346][say 359][yey 296]...
[mwu 114][twu 173][phwu 187][pwu 239][tok 277][mok 280][lwu 285][mo 307]...
[son 122][sun 168][chon 327][con 341][un 363][ton 436][lon 475][nwun 475]...
[so 120][o 445][su 453][u 520][ssu 578][no 692][swu 745][hwu 782]...
[li 163][la 172][le 232][toi 241][kwui 281][hi 286][hoi 299][mek 303]...
[ka 34][ki 95][ke 302][khi 302][ca 320][ci 352][kye 489][kki 491]...
[tul 87][tut 105][thul 160][mwul 197][tum 218][tol 219][lul 254][nul 314]...
[lye 106][le 162][the 200][mek 203][li 254][hye 255][chye 270][phe 280]...
[noph 202][nwun 266][noh 287][nol 297][lyo 304][pon 318][tal 362][mok 370]...
[tol 187][ul 210][mwul 225][sol 241][nol 267][phwul 271][dwul 275][swul 283]...
[tte 211][tta 320][ppe 488][mye 516][payk 534][tey 551][ye 558][me 577]...
[po 84][mo 282][o 284][pwu 315][u 333][yo 385][mu 395][pok 417]...
[ni 66][na 145][si 334][ne 349][sa 364][i 387][nye 409][a 455]...
[yeph 124][iph 182][ilh 197][anh 282][el 283][aph 288][teph 292][et 296]...
[ip 248][cip 251][ip 286][cap 293][yeng 294][cam 299][cing 302][ching 309]...
.....
.....
```

Figure 2: Output of the recognition device for the example sentence: *camkyel-ey mwusun soli-ka tul-lye nwun-ul tte po-ni yepcip-i pwul-ey tha-ko issess-ta* (When I opened my eyes by overhearing something asleep, the neighboring house was in flames).

[cim 104][cam 137]

[kyel 31]

[ney 181][ey 186][sey 232][ay 245]

[mwu 114][twu 173]

[son 122][sun 168]

[so 120]

[li 163][la 172][le 232][toi 241]

[ka 34]

[tul 87][tut 105][twul 87]

[lye 106]

[noph 202][nwun 266][noh 287][nol 297][lyo 304]

[tol 187][ul 210][mwul 225][sol 241][nol 267][phwul 271][dwul 275][swul 283]...

[tte 211][tta 320]

[po 84]

[ni 66]

[yeph 124][iph 182]

[ip 248][cip 251][ep 286][cap 293][yeng 294][cam 299][cing 302][ching 309]...

[a 191][i 207][e 267][ya 287]

[pwul 61][mwul 78]

[ney 284][ey 315][sey 382][ay 388]

[tha 138]

[ko 132]

[iss 168][ass 175][ess 201]

[ess 127][ass 161][iss 184][es 185]

[ma 227][ne 318][na 332][ta 277]

Figure 3: The candidate selection result

```
T[i,n] <- find_word_from_trie_dict, 1<=i<=n; /* fill last column */
for (start = n; start >= 1; start --) {
   if (!Empty (T[start,n])) {
     T[i,start-1] <- find_word_from_trie_dict, 1<=i<=start-1;


     for (left_start = start -1 ; left_start >= 1 ; left_start --) {
        /* now begin connectivity checking */
       foreach left_morph_chain in T[left_start,start-1]  {
         /* more than one chain */
        foreach right_morph_chain in T[start,n] {
         if Connectable (left_morph_chain, right_morph_chain)
          AddTo(T[left_start,n], concat(left_morph_chain,right_morph_chain));
        } /* for right morpheme */
       } /* for left morpheme */
     } /* for left_start */
   } /* if */
}  /* for start */
```

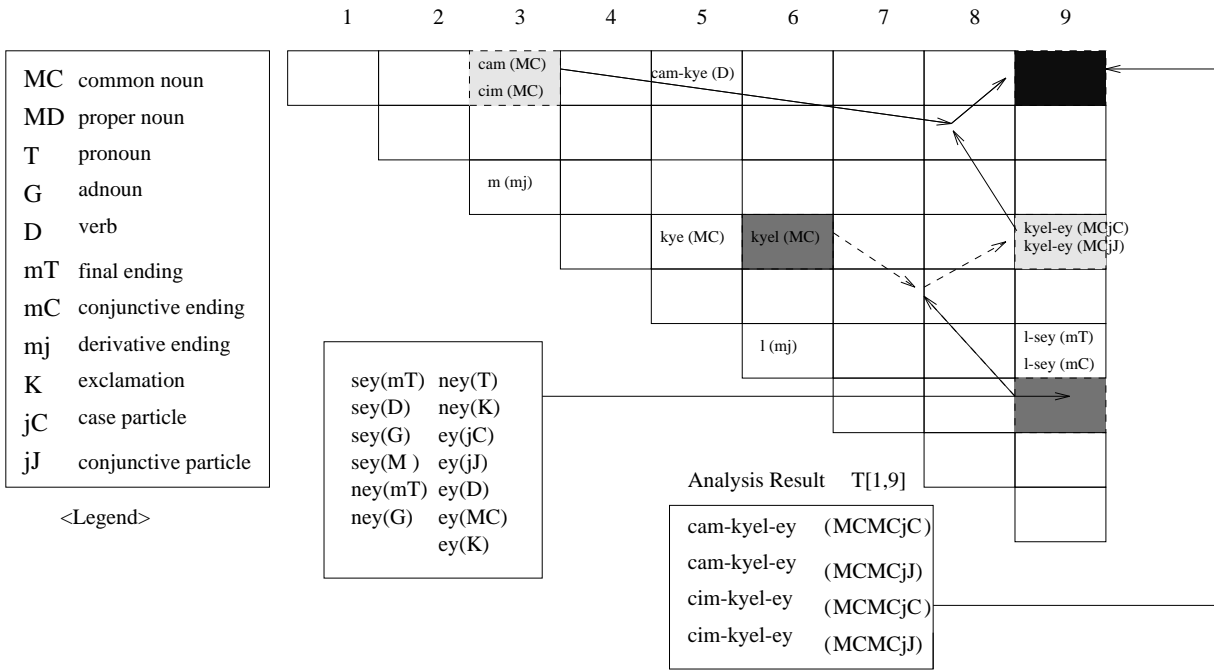Figure 4: Morphological analysis algorithm based on the tabular parsing method

	1	2	3	4	5	6	7	8	9
			cam (MC) cim (MC)		cam-kye (D)				
		m (mj)							
				kye (MC)	kyel (MC)			kyel-ey (MCjC) kyel-ey (MCjJ)	
					l (mj)			l-sey (mT) l-sey (mC)	

MC common noun
MD proper noun
T pronoun
G adnoun
D verb
mT final ending
mC conjunctive ending
mj derivative ending
K exclamation
jC case particle
jJ conjunctive particle

<Legend>

sey(mT) ney(T)
sey(D)   ney(K)
sey(G)   ey(jC)
sey(M) ey(jJ)
ney(mT) ey(D)
ney(G)   ey(MC)
         ey(K)

Analysis Result   T[1,9]

cam-kyel-ey	(MCMCjC)
cam-kyel-ey	(MCMCjJ)
cim-kyel-ey	(MCMCjC)
cim-kyel-ey	(MCMCjJ)

Figure 5: Morphological analysis results for the first eojeol *camkyel-ey* in the recognized sentence. Among the total 8 *eojeol* candidates, only the morphologically correct sequences (with each 2 different POS tags) remain in the final position T[1,9].
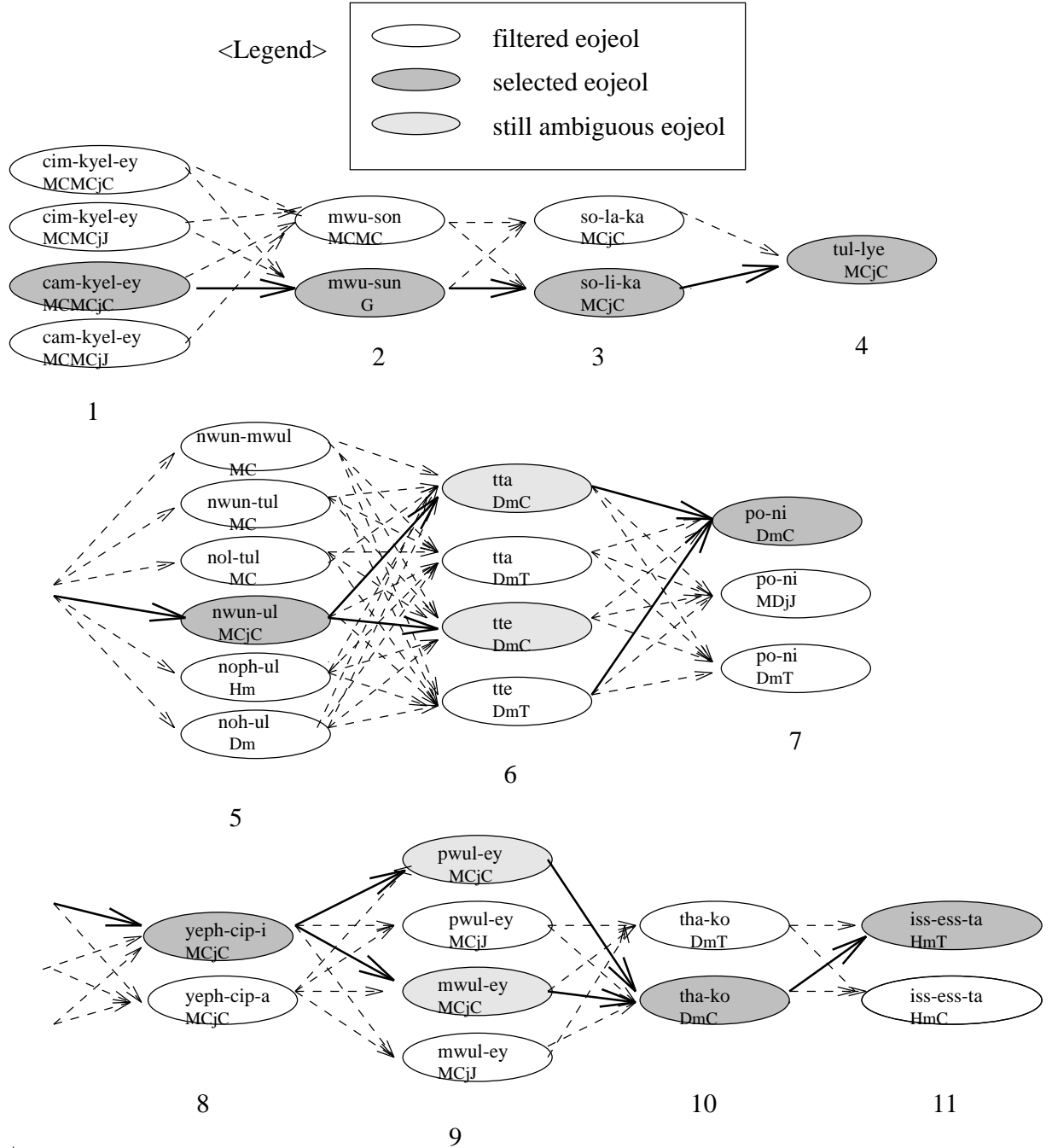
Figure 6: The ambiguity resolution using the tri-gram tagging after morphological analysis. The numbers represent the *eojeol* positions in the sentence.
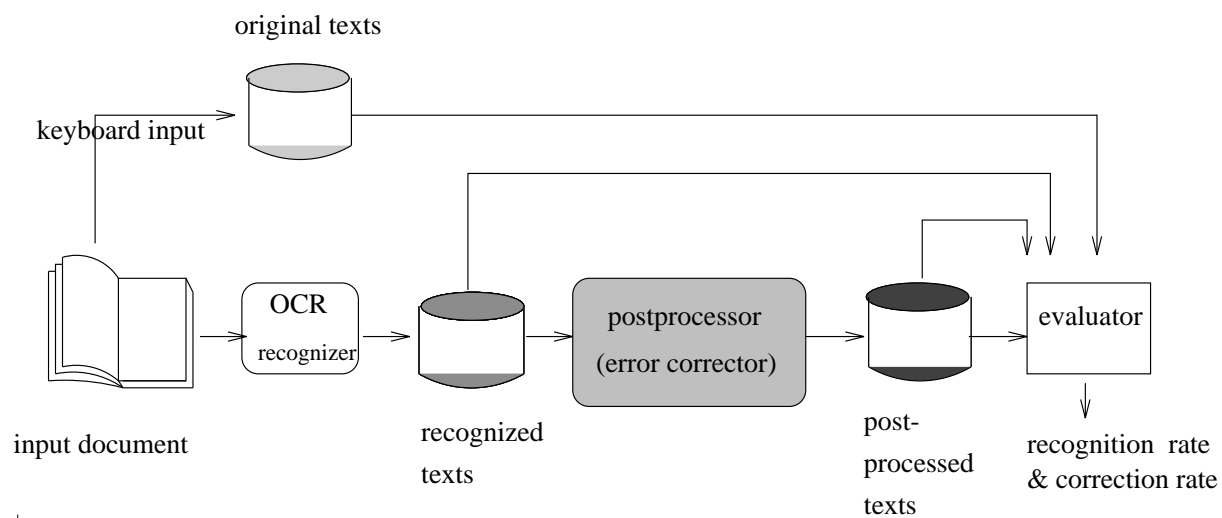
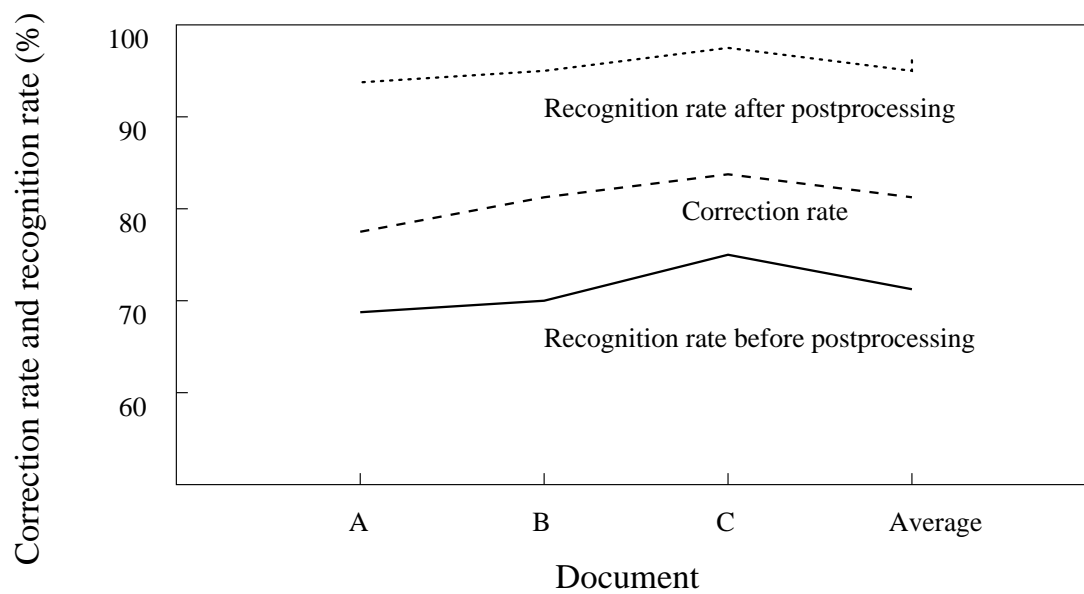Figure 7: The experiment set-up for the multi-level post-processing

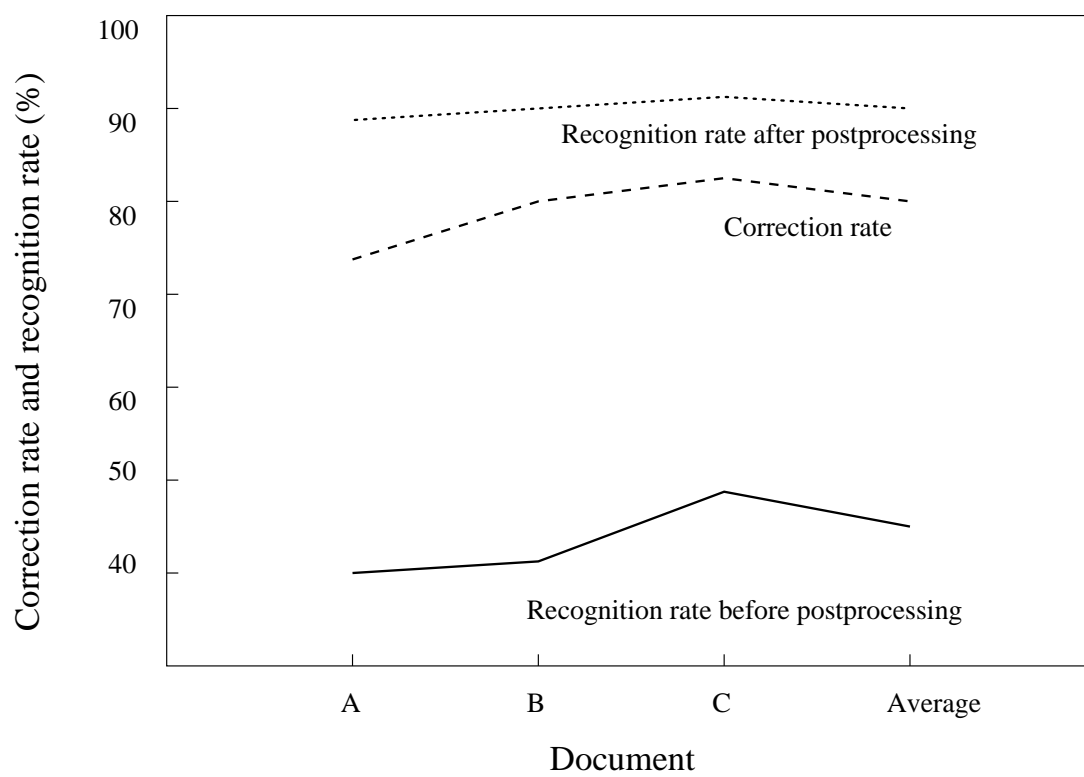Figure 8: Recognition rate and correction rate for characters (with each test data-set A, B, C)

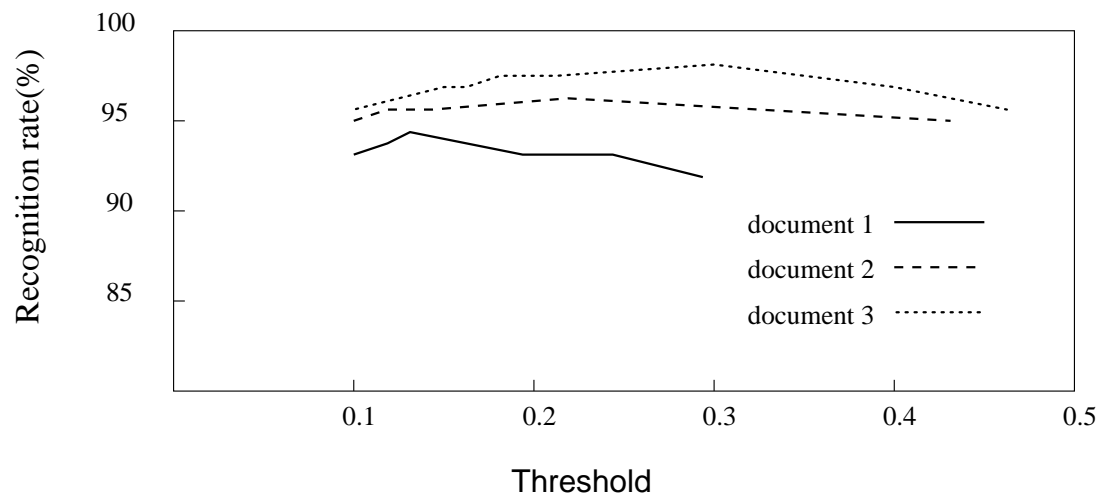Figure 9: Recognition rate and correction rate for *eojeols*

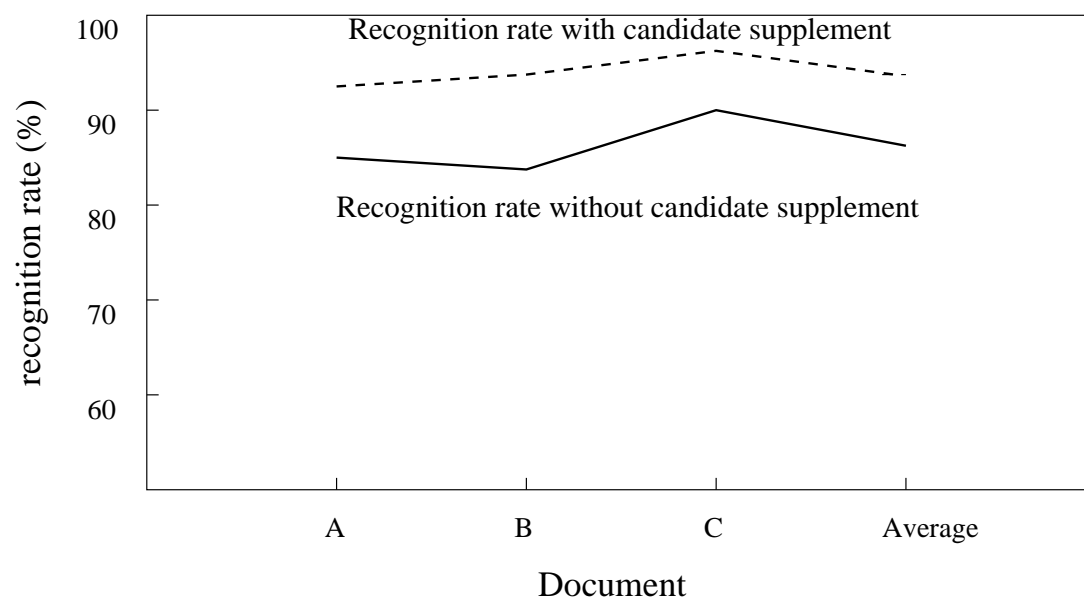Figure 10: Candidate restriction threshold and recognition rate

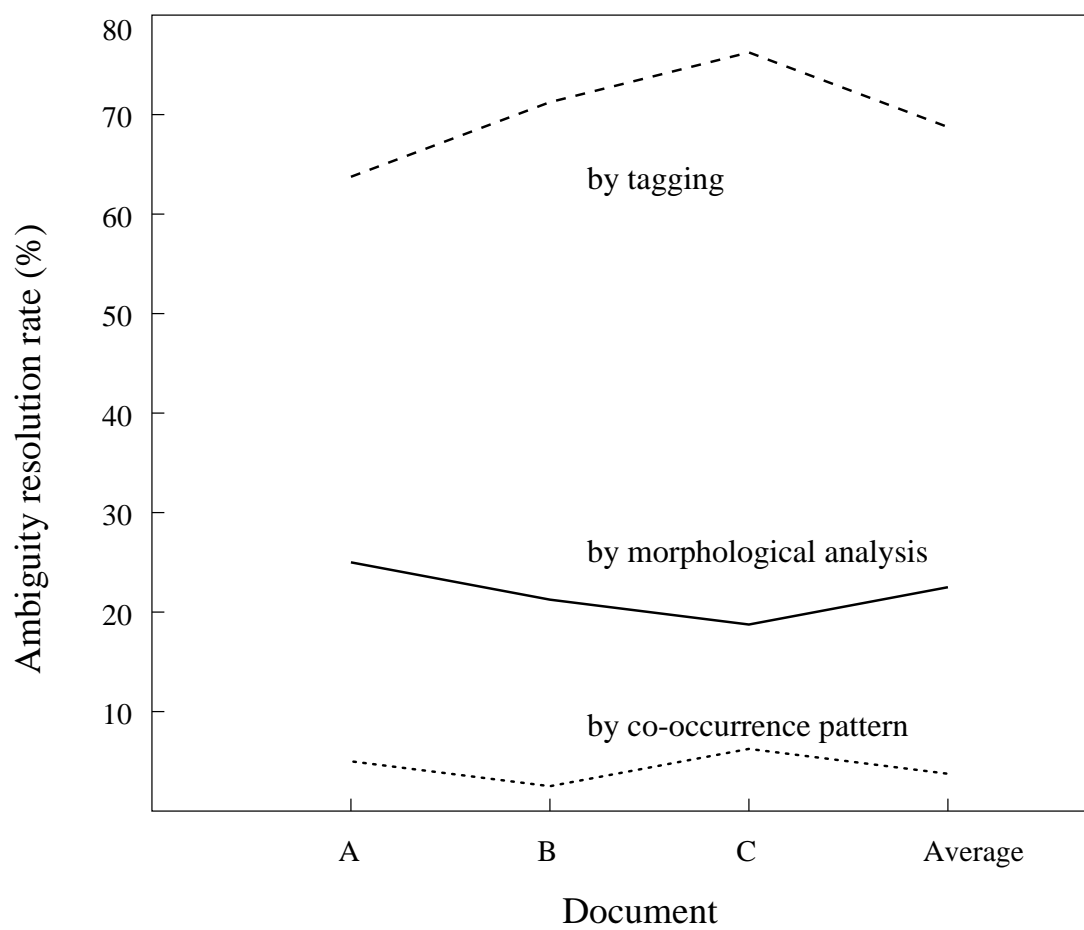Figure 11: The candidate supplement effects using similar character sets

Figure 12: Disambiguation performance of each linguistic constraint

Biographical sketches for each author

Geunbae Lee received his B.S. and M.S. degrees in computer engineering from the Seoul National University, Seoul, Korea in 1984 and 1986, respectively and the Ph.D. degree in computer science from the University of California at Los Angeles in 1991. Currently, he is an assistant professor in the Computer Science Department, Pohang University of Science and Technology, Pohang, Korea. His current research interests are in the field of natural language processing, speech recognition, and intelligent agent interfaces.

Jong-Hyeok Lee received his B.S. degree in mathematics eduation from the Seoul National University, Seoul, Korea in 1980, and his M.S. and Ph.D degree in computer science from KAIST, Korea in 1982 and 1988, respectively. Currently, he is an assistant professor in the Computer Science Department, Pohang University of Science and Technology, Pohang, Korea. His current research interests are in the field of natural language processing, machine translation, Korean information processing.

JinHee Yoo received her B.S. and M.S. degrees in computer science from the Pohang University of Science and Technology, Pohang, Korea in 1993 and 1995, respectively. Currently she works for SamSung Electronics Company. Her current research interests are in the field of natural language processing and character recognition.